Decentralized K-means using randomized Gossip protocols for clustering large datasets

Jerome Fellus*, David Picard* and Philippe-Henri Gosselin†
jerome.fellus@ensea.fr, picard@ensea.fr, gosselin@ensea.fr
*ETIS, UMR 8051, ENSEA/Universite de Cergy-Pontoise/CNRS, 6 Avenue du Ponceau 95014 Cergy, France
†INRIA, Rennes Bretagne Atlantique, France

***Abstract*—In this paper, we consider the clustering of very large datasets distributed over a network of computational units using a decentralized K-means algorithm. To obtain the same codebook at each node of the network, we use a randomized gossip aggregation protocol where only small messages are exchanged. We theoretically show the equivalence of the algorithm with a centralized K-means, provided a bound on the number of messages each node has to send is met. We provide experiments showing that the consensus is reached for a number of messages consistent with the bound, but also for a smaller number of messages, albeit with a less smooth evolution of the objective function.**

I. Introduction

Unsupervised clustering is a popular learning task with many application fields such as data compression, computer vision, and data mining. Available contents in those fields are growing exponentially and with no doubt faster than the computing performances of individual machines. Besides, data tends to originate more and more often from decentralized sources (mobile devices, wireless sensors,...). Such trends raise a challenge for new clustering techniques which have to cope with sizes and layouts of real world data.

Given a set $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ of n samples in $\mathbb{R}^D$, we are interested in producing a *codebook* $\mathcal{M} = \{\mu_1 \dots \mu_K\}$ made of K vectors in $\mathbb{R}^D$ called *codewords* or *centroids*. $\mathcal{M}$ determines a Voronoi partition of $\mathbb{R}^D$ in K *clusters* $\{\mathcal{C}_k\}$ and a quantization function $q : \mathbb{R}^D \rightarrow \{1..K\}$ so that $\mathcal{C}_k = \{\mathbf{x} \ / \ q(\mathbf{x}) = k\}$. Our goal is to compute a codebook which minimizes the intra-cluster mean square error (MSE):

$$\text{MSE}(\mathcal{M}) = \frac{1}{N} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \mu_{q(\mathbf{x}_i)} \right\|_2^2 \tag{1}$$

Unfortunately this problem is NP-hard [1]. Most approaches to solve it adopt incremental optimization strategies [2], but assume that all data is available at a single location. When the whole data is spread over a network of processing units, a distributed optimization problem is raised.

In this paper we investigate decentralized K-means clustering, where N processing nodes shipped with their own local datasets $\mathbf{X}^{(1)} \dots \mathbf{X}^{(N)} \subset \mathbb{R}^D$ have to jointly optimize local codebooks $\mathcal{M}^{(i)} = \{\mu_k^{(i)}\}, 1 \leq i \leq N$ so as to model the whole data $\mathbf{X} = \bigcup_i \mathbf{X}^{(i)}$. After introducing centralized K-means, distributed strategies and gossip-based aggregation in section 2, we propose in section 3 a decentralized K-means algorithm based on randomized gossip protocols with a decentralized codeword-shifting feature. In section 4, We prove that it yields codebooks $\mathcal{M}^{(i)}$ which are both consistent over all nodes and equivalent to centralized approaches in terms of MSE, while ensuring a bound on the number of exchanged messages. Section 5 is devoted to its experimental analysis on both synthetic and large real datasets. We conclude the paper giving future research directions in section 6.

II. Related Work

In this section, we present an overview of centralized and distributed K-means algorithms. We show that distributed K-means can be efficiently solved using decentralized weighted averaging techniques. Finally, we introduce Gossip aggregation protocols to perform this averaging.

A. Centralized approaches

The first proposals for solving K-means [2] are still widely used. Their basic idea is to iteratively improve a codebook $\mathcal{M}(\tau)$ with a gradient descent strategy. Starting from a random codebook $\mathcal{M}(0)$, two optimization steps are performed at each iteration τ:

Step 1 : Clusters assignment. Given $\mathcal{M}(\tau)$, compute the clusters $\{\mathcal{C}_k\}(\tau)$ by mapping each sample $\mathbf{x} \in \mathbf{X}$ to its nearest neighbor in $\mathcal{M}(\tau)$:

$$\forall k, \quad \mathcal{C}_k(\tau) = \{\mathbf{x} \in \mathbf{X} \ / \ \underset{\mu_k(\tau) \in \mathcal{M}(\tau)}{\arg\min} \ \|\mathbf{x} - \mu_k(\tau)\|_2^2\} \tag{2}$$

Step 2 : Codebook update. Given the clusters $\{\mathcal{C}_k\}(\tau)$, compute a new codebook $\mathcal{M}(\tau + 1)$ as their barycenters :

$$\forall k, \quad \mu_k(\tau + 1) = \frac{1}{|\mathcal{C}_k(\tau)|} \sum_{\mathbf{x}_i \in \mathcal{C}_k(\tau)} \mathbf{x}_i \tag{3}$$

Alternating clusters assignment and codebook update, $\mathcal{M}(\tau)$ is shown to reach a local minimum of the MSE. To escape from local minima with empty or unbalanced clusters, codeword-shifting techniques that re-assign low density clusters to higher density regions have been proposed [3], [4].

Note that if step 2 requires $\mathcal{O}(K)$ vector sums computations, step 1 requires $\mathcal{O}(nK)$ distances computations, which can be very costly when coupling large datasets with large codebooks. To deal with such situations, several approaches were proposed to overcome the need for random access to data, resulting in so-called *online* methods (stochastic VQ, streaming [5], [6], and others based on competitive learning [7]). Still, these techniques rely on a single processing unit. They remain quite time-consuming or make intensive use of approximate computations [6].

B. Distributed clustering strategies

To follow a hierarchy by Bandyopadhyay et al. [8], distributed clustering schemes can be grouped in two main categories:

One round models combination. In these approaches, each node i first computes a local model of its own data using a centralized clustering technique. A global combination is then made in only one communication round, either transmitting compact representations of those local models to a central site [9] or propagating them between neighboring nodes [10]. The number of exchanged messages is generally small, but the global model quality strongly depends on the generalization ability of the local models, which is often difficult to obtain.

Incremental aggregations. In contrast, incremental approaches perform a global aggregation step after each local K-means optimization step. They take benefit from the intrinsic data parallelism of the assignment step [11]–[14]. Each node performs assignment (step 1) independently on its own data, all samples being assigned to one cluster. This yields a global parition of $\mathbf{X}$ in K clusters. The associated optimal codebook $\mathcal{M}^\star(\tau)$, called *consensus codebook*, is then made of their barycenters $\{\mu_k^\star\}$.

$\mathcal{M}^\star(\tau)$ can not be locally computed at node i, since it involves the locally unknown global partition. Still, computing the barycenters $\mu_k^{(i)}$ of the locally computed clusters $\mathcal{C}_k^{(i)}$ forms local codebooks $\mathcal{M}^{(i)} = \{\mu_1^{(i)} \dots \mu_K^{(i)}\} \equiv [\mu_1^{(i)} \dots \mu_K^{(i)}]$ which optimally fit local data. It is easy to see that the *consensus codewords* $\mu_k^\star$ are actually the barycenters of the $\mu_k^{(i)}$ computed by all nodes i, weighted by the corresponding clusters sizes $n_k^{(i)} = |\mathcal{C}_k^{(i)}|$:

$$\mu_k^\star = \frac{\sum_i \sum \{\mathbf{x} \in \mathcal{C}_k^{(i)}\}}{\sum_i n_k^{(i)}} = \frac{\sum_i n_k^{(i)} \mu_k^{(i)}}{\sum_i n_k^{(i)}} \,. \tag{4}$$

Therefore, estimating $\mathcal{M}^\star$ amounts to compute an entry-wise weighted average of all the $\mathcal{M}^{(i)}$ taken as matrices $[\mu_k^{(i)}]_k$ in a decentralized fashion. A large work is available on distributed averages computation, including transmission to a central site [15], combination along a spanning tree (e.g. using an Echo/Probe walk on the network [16]) and local neighboring nodes random sampling [17]. Centralized aggregation suffers from the obvious communication bottleneck at the master node. Hierarchical aggregation is bandwidth-friendly, but very sensitive to nodes failures. Neighborhood sampling exhibits limited and balanced communication costs as well as potential asynchrony, but unbiased estimates and diffusion speeds can not be guaranteed.

C. Decentralized averaging using Gossip protocols

Decentralized averaging consists in estimating the average of N values locally hosted by N nodes only using local exchanges between neighboring nodes. Each node i holds a local value $v_i(0) \in \mathbb{R}$, and must converge to $\bar{v} = \frac{1}{N}\sum_i v_i(0)$. Decentralized averaging can be solved using Gossip-based protocols [18]. Gossip protocols exhibit low communication costs, robustness to failure and scalability to large networks with point-to-point connectivity. Each node i regularly sends a fraction $\kappa_{ij}(t) \geq 0$ of its current estimate $v_i(t)$ to a randomly selected set of neighboring nodes and keeps the remaining portion. The absence of message from i to j is rendered by $\kappa_{ij}(t) = 0$. Upon receipt of a value v, the target node j adds it to its local $v_j(t)$. The coefficients $\kappa_{ij}(t)$ are randomly sampled from a predefined random distribution and must respect a so-called *mass-conservation* constraint: $\forall t, i, \sum_j \kappa_{ij}(t) = 1$. Noting the vector $\mathbf{v}(t) = (v_1(t) \dots v_N(t))^T$, we get a corresponding matrix form:

$$\mathbf{v}(t+1)^T = \mathbf{v}(t)^T \mathbf{K}(t), \quad \text{with } \mathbf{K}(t) = [\kappa_{ij}(t)]_{ij} \tag{5}$$

The distribution of the *diffusion matrices* $\mathbf{K}(t)$ fully characterizes a given protocol. Mass conservation implies row-stochastic diffusion matrices (*i.e.* $\mathbf{K}(t)\mathbf{1} = \mathbf{1}$) and entails that the sum of all node values is kept constant. Then, provided that nodes converge to a same value, this value will be $\bar{v}$.

$$\forall t, \mathbf{v}(t+1)\mathbf{1} = \mathbf{v}(t)^T \mathbf{K}(t)\mathbf{1} = \mathbf{v}(t)^T \mathbf{1}$$
$$\forall t, \textstyle\sum_i v_i(t) = \sum_i v_i(0) = N\bar{v}$$

However, convergence requires such a state to be a fixed-point of the diffusion matrices ($\bar{v}\mathbf{1}^T\mathbf{K}(t) = \bar{v}\mathbf{1}^T$, *i.e*, $\mathbf{K}(t)$ are also column-stochastic). Any message from i to j must then come with a (simultaneous) message from j to i [19]. This can be problematic in unreliable routing environments.

Sum-Weight protocols [18], [20], [21] were proposed to release the column-stochasticity constraint, thus allowing one-way *push* messages. They couple each estimate $v_i(t)$ with a weight $w_i(t)$ updated with the same rules ($w_i(0) = 1$). The quotients $v_i(t)/w_i(t)$ are shown to be consistent estimators for $\bar{v}$. Interestingly, weighted averages can be computed using initial weights other than 1. Gossip protocols were successfully applied to decentralized K-means clustering [11], [12] and decentralized Expectation Maximization (EM) on Gaussian Mixtures Models [10].

In this paper, we extend the work on Gossip K-means from [11]. We prove it to be equivalent to the centralized version (*i.e*, the partitions obtained at each K-means iteration are the same), provided that a sufficient condition on the number of exchanged messages is met. Furthermore, we improve the output codebook by including a decentralized codeword shifting procedure.

III. Randomized Gossip clustering algorithm

Our proposed Gossip K-means algorithm is presented in Algorithm 1. We describe the procedure for a node i, all nodes running the same procedure independently and starting with an initial local codebook $\mathcal{M}^{(i)}(0)$ independently drawn at random. At each iteration τ, i first performs a local optimization step on its own data, and then estimates the consensus codebook $\mathcal{M}^\star(\tau)$ using a randomized Gossip aggregation protocol.

A. Local optimization (step 1)

Each local sample $\mathbf{x} \in \mathbf{X}^{(i)}$ is assigned the cluster $\mathcal{C}_k^{(i)}(\tau)$ associated with the nearest codeword in the current local codebook. Meanwhile, we compute the size, sum and squared error for each cluster $\mathcal{C}_k^{(i)}(\tau)$, (denoted by $n_k^{(i)}(\tau)$, $\mathbf{s}_k^{(i)}(\tau)$ and $d_k^{(i)}(\tau)$). The locally optimal codebook is thus $\mathbf{s}_k^{(i)}(\tau)/n_k^{(i)}(\tau)$.

Algorithm 1: Gossip K-Means with codeword-shifting

At node i
Input:
$\mathbf{X}^{(i)} = \{\mathbf{x}^{(i)}\} \subset \mathbf{X}$: Local training dataset for node i
K : Number of desired clusters
Δ_{min} : MSE improvement convergence criterion
σ_{shift} : Clusters errors RSD termination threshold
Output: A local codebook $\mathcal{M} = [\mu_k]_k$
begin
  Initialize $\mathcal{M} = [\mu_k]_k$ at random ; $\text{MSE}_{old} \leftarrow \infty$;
  repeat
    $(n_k)_k \leftarrow \mathbf{0}_K$; $[\mathbf{s}_k]_k \leftarrow \mathbf{0}_{K\times D}$; $(d_k)_k \leftarrow \mathbf{0}_K$
    foreach $\mathbf{x} \in \mathbf{X}_i$ **do**
      $k \leftarrow \arg\min_{k'} \|\mathbf{x} - \mu_{k'}\|_2^2$
      n_k+= 1 ; $\mathbf{s}_k$+= $\mathbf{x}$; d_k+= $\|\mathbf{x} - \mu_k\|_2^2$
    end
    Run **EmissionProcedure()** and **ReceptionProcedure()** concurrently
    $\mathcal{M} \leftarrow [\mathbf{s}_k/w_k]_k$; $(d_k)_k \leftarrow (d_k/w_{K+1})_k$;
    $\text{MSE} \leftarrow \frac{1}{K}\sum_k d_k$
  while CodewordShifting()
end

Algorithm 2: EmissionProcedure

$\mathbf{w} \leftarrow (n_1, \ldots, n_k, 1)$
for $t \leftarrow 1$ **to** M **do**
  Draw a neighbor node $j \in \mathcal{N}_i$ at random
  $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w}) \leftarrow \frac{1}{2}([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w})$
  Send $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w})$ to j
end

Algorithm 3: ReceptionProcedure

repeat
  Upon receipt of a message $([\mathbf{s}_k^{(j)}]_k, (d_k^{(j)})_k, \mathbf{w}^{(j)})$
  $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w})$ += $([\mathbf{s}_k^{(j)}]_k, (d_k^{(j)})_k, \mathbf{w}^{(j)})$
while *EmissionProcedure is running*

B. Gossip aggregation (step 2)

Step 1 yields a global partition $\{\bigcup_j \mathcal{C}_1^{(j)}, \ldots, \bigcup_j \mathcal{C}_K^{(j)}\}$ of $\mathbf{X}$ which has an associated optimal codebook $\mathcal{M}^\star(\tau)$ made of the locally unknown barycenters $\mu_k^\star(\tau)$ of the $\bigcup_j \mathcal{C}_k^{(j)}$ (Eq. 4).

$$\forall k, \quad \mu_k^\star(\tau) = \frac{1}{\sum_i n_k^{(i)}(\tau)} \sum_i \mathbf{s}_k^{(i)}(\tau) \tag{6}$$

These weighted averages are estimated using a dedicated asynchronous sum-weight gossip protocol. Each node i maintains a $D\times K$ estimates matrix $\mathbf{S}^{(i)} = [\mathbf{s}_1^{(i)} \ldots \mathbf{s}_K^{(i)}]$ and a K-dimensional squared errors vector $\mathbf{d}^{(i)}$. Each $\mathbf{s}_k^{(i)}$ is coupled with a weight $w_k^{(i)}$, $\mathbf{d}^{(i)}$ being coupled with $w_{K+1}^{(i)}$. Starting with $\mathbf{S}^{(i)} = [\mathbf{s}_1^{(i)}(\tau) \ldots \mathbf{s}_K^{(i)}(\tau)]$, $\mathbf{d}^{(i)} = (d_1^{(i)}(\tau) \ldots d_K^{(i)}(\tau))^T$ and $\mathbf{w}^{(i)} = (n_1^{(i)}(\tau) \ldots n_K^{(i)}(\tau), 1)^T$, each node i runs emission and reception procedures concurrently:

Emission procedure. Repeatedly, node i divides all entries of $\mathbf{S}^{(i)}$,$\mathbf{d}^{(i)}$ and $\mathbf{w}^{(i)}$ by 2 and sends them to a randomly chosen neighboring node j.

Reception procedure. Upon receipt of a message from node j, i adds received entries to the corresponding ones in its own $\mathbf{S}^{(i)}$, $\mathbf{d}^{(i)}$ and $\mathbf{w}^{(i)}$.

After sending M messages, i obtains local estimates of the consensus codewords and the average cluster-wise squared errors by dividing the entries of $\mathbf{S}^{(i)}$ and $\mathbf{d}^{(i)}$ by their associated weights in $\mathbf{w}^{(i)}$. Using notations from section II.C, a message from i to j boils down to applying (Eq. 5) to each entry of $(\mathbf{S}^{(i)}, \mathbf{d}^{(i)}, \mathbf{w}^{(i)})$ with:

$$\mathbf{K}(t) = \mathbf{I} + \frac{1}{2}\mathbf{e}_i \left(\mathbf{e}_j - \mathbf{e}_i\right)^T \tag{7}$$

where i uniformly drawn at random in $\{1 \ldots N\}$
j uniformly drawn at random in $\{1 \ldots N\} \setminus i$

The t^{nth} message exchanged at iteration τ is thus represented by the random matrix $\mathbf{K}(t)$ which only involves one sender i and one receiver $j \neq i$. In section 4, we derive a minimal value for M, above which the codeword estimates become sufficiently close to the consensus ones to ensure that our algorithm produces partitions that are equivalent to a centralized K-means at each iteration τ.

C. Codeword-shifting and termination test (step 3)

Algorithm 4: CodewordShifting

if $0 \leq \frac{MSE_{old} - MSE}{MSE_{old}} < \Delta_{min}$ **then**
  if $\frac{\sqrt{\sum_k (d_k - MSE)^2}}{MSE} < \sigma_{shift}$ **then return** *False*;
  else
    $k_{max} = \arg\max_k d_k$; $k_{min} = \arg\min_k d_k$;
    $\mu_{k_{min}} \leftarrow \mu_{k_{max}} + \eta$; $\mu_{k_{max}} \leftarrow \mu_{k_{max}} - \eta$;
  end
end
$\text{MSE}_{old} \leftarrow \text{MSE}$;
return *True*

Our codeword shifting feature is a node-local procedure inspired from the centralized approach of [4]. It is triggered when the relative improvement of the global MSE is lower than a given threshold Δ_{min} (*i.e*, the MSE converged to a minimum) but the relative standard deviation (RSD) of the clusters squared errors is higher than a given σ_{shift} (*i.e*, the clusters are unbalanced). The MSE is locally estimated using the squared errors estimates $\mathbf{d}^{(i)}$ from step 2.

$$\text{RSD}^{(i)} = \sqrt{\sum_{k=1}^{K} \left(d_k^{(i)} - \text{MSE}^{(i)}\right)^2} \quad \text{MSE}^{(i)} = \frac{1}{K}\sum_{k=1}^{K} d_k^{(i)}$$

If node i satisfies the codeword shifting condition, it shifts the codeword with lowest squared error k_{min} near the one with highest squared error k_{max}. More precisely, $\mu_{k_{min}}$ becomes $\mu_{k_{max}}$ plus a small arbitrary vector η. Although this selection criterion is weaker than [4], empty clusters are progressively avoided and the produced codebooks have better balanced clusters squared errors, leading to a more fair density modeling.

If the RSD is lower than a threshold σ_{shift}, the algorithm terminates at node i and $\mathcal{M}^{(i)}(\tau+1)$ is returned.

IV. Convergence analysis

Here we analyze the convergence of our algorithm and derive a probabilistic bound on the number of messages M each node i has to send at each iteration. This bound ensures that the error resulting from the decentralized aggregation step will be low enough to have no impact on subsequent operations, *i.e*, the algorithm's behavior at each iteration τ is equivalent to a centralized K-means trained on the whole data. Firstly, we identify a criterion to evaluate this equivalence, and then we find the number of messages exchanges required to reach this criterion. The full proofs of the needed lemmas and theorems are deferred to Appendix A.

To ensure that the algorithm behaves the same as a centralized K-means, a sufficient condition is to get the same global partition of $\mathbf{X}$ as the centralized version at each iteration. This yields a so-called *zero-mismatch* constraint on the local codebooks. Focusing on a single iteration, we omit τ for better readability. Denoting by $\hat{\mu}_1^{(i)} \dots \hat{\mu}_K^{(i)}$ the codewords estimates at node i and by $\mu_k^\star$ the consensus codewords, zero-mismatch corresponds to satisfying:

$$\forall i, \forall \mathbf{x} \in \mathbf{X}^{(i)}, \arg\min_k \left\| \mathbf{x} - \hat{\mu}_k^{(i)} \right\|_2 = \arg\min_k \left\| \mathbf{x} - \mu_k^\star \right\|_2 .$$

Using simple triangular inequalities, and denoting by $m_{\text{NN}}^{(i)}(\mathbf{x})$ the m^{nth} nearest neighbor of $\mathbf{x}$ among the $\{\hat{\mu}_k^{(i)}\}$, the following lemma gives an upper bound on the estimates squared error to meet the zero-mismatch constraint:

Lemma IV.1 *The zero-mismatch constraint is met if*

$$\forall i, \quad \left\| \mathcal{M}^{(i)} - \mathcal{M}^\star \right\|_F^2 \leq K \epsilon_{zmc} \quad \textit{where}$$
$$\epsilon_{zmc} = \frac{1}{2} \min_i \min_{\mathbf{x} \in \mathbf{X}_i} \left(\left\| \mathbf{x} - 2_{NN}^{(i)}(\mathbf{x}) \right\|_2^2 - \left\| \mathbf{x} - 1_{NN}^{(i)}(\mathbf{x}) \right\|_2^2 \right) .$$

We now need to find the number of messages to exchange in order to reach this error bound for all local codebook estimates. We adopt the same approach as in [21]. $\mathbf{I}$ and $\mathbf{1}$ respectively denote the $N \times N$ identity matrix and the N-dimensional all ones vector. Our protocol being driven by the diffusion matrices $\mathbf{K}(t)$ from Eq.7, and noting $\mathbf{P}(t) = \mathbf{K}(1)\mathbf{K}(2)\dots\mathbf{K}(t)$, we first state this lemma:

Lemma IV.2 *The estimation error at any node i after t exchanged messages is always upper-bounded as follows:*

$$\forall t, i, \quad \left\| \hat{\mathcal{M}}^{(i)}(t) - \mathcal{M}^\star \right\|_F^2 \leq KD \max_k \frac{\sum_i w_k^{(i)}(0)^2}{\min_i w_k^{(i)}(t)^2} \psi(t)$$
$$\textit{where} \quad \psi(t) = \max_k \|(\mathbf{I} - \tilde{\mathbf{w}}_k \mathbf{1}^T)\mathbf{P}(t)\|_F^2$$
$$\textit{with} \quad \tilde{\mathbf{w}}_k = \frac{1}{\sum_i^N w_k^{(i)}(0)} (w_k^{(1)}(0) \dots w_k^{(N)}(0))$$

This bound involves two terms which depend on t. We thus upper-bound $\psi(t)$ and lower-bound $\min_j w_k^{(j)}(t)$. Using $\mathbf{P}(t+1) = \mathbf{P}(t)\mathbf{K}(t+1)$, $\psi(t)$ is upper-bounded by taking its conditional expectation given $\psi(t-1)$ to get an expression of $\mathbb{E}[\psi(t)]$. Markov's inequality then yields an upper bound on $\psi(t)$ with probability greater than δ:

Lemma IV.3 *$\psi(t)$ is upper-bounded as follows:*

$$\forall \delta, \forall \epsilon, \forall t \geq \frac{\ln N - \ln(1-\delta) - \ln \epsilon}{\ln \frac{2N^2-2N}{2N^2-3N-1}} \quad \mathbb{P}[\psi(t) \leq \epsilon] \geq \delta$$

To lower-bound $\min_i w_k^{(i)}(t)$ we observe that there is always a weight which is greater than the average $\bar{w}_k$ of all nodes weights for cluster k and analyze its diffusion speed. That is, we find the minimum number of messages T_δ beyond which all nodes received a part of this weight with probability at least δ. As no weight can decrease by more than half its value at each message event, weights are never lower than $\bar{w}_k 2^{-T_\delta}$.

Lemma IV.4 *$\min_i w_k^{(i)}(t)$ is lower bounded as follows:*

$$\forall \delta, \forall t > T_\delta, \forall k, \quad \mathbb{P}[\min_i w_k^{(i)}(t) \geq \bar{w}_k 2^{-T_\delta}] \geq \delta$$

The analytical expression of T_δ is difficult to obtain, but a lookup table using the inverse function can be easily computed and is given in Appendix B.

Gathering the results of Lemmas IV.1-4, the following theorem expresses the total number of messages $\mathcal{T}$ required to meet the zero-mismatch constraint:

Theorem IV.5 *The zero-mismatch constraint is met with probability δ when at least $\mathcal{T}$ messages have been exchanged in the network, where*

$$\mathcal{T} = \frac{3\ln N + 2T_{\delta^{1/2}} \ln 2 + \ln D - \ln(1-\delta^{1/2}) - \ln \epsilon_{zmc}}{\ln \frac{2N^2-2N}{2N^2-3N-1}}$$

To have $\mathcal{T}$ exchanges, it is sufficient that each node send at least $M = \mathcal{T}/N$ messages. We then obtain the number of messages a single node has to send to ensure equivalence with a centralized K-means algorithm:

Theorem IV.6 *An iteration of the distributed algorithm given in Algorithm 1 is equivalent to a centralized K-means iteration with probability greater than δ provided that every node i sends at least M messages, with*

$$M = \frac{3\ln N + 2T_{\delta^{1/2}} \ln 2 + \ln D - \ln(1-\delta^{1/2}) - \ln \epsilon_{zmc}}{N \ln \frac{2N^2-2N}{2N^2-3N-1}}$$

V. Experiments

The proposed algorithm has been evaluated both on synthetic and real-world data, using various indicators from the decentralized clustering literature [8]. The MSE over the whole data (MSE_g) evaluates modeling and generalization ability of the decentralized algorithm at each node.

$$\text{MSE}_g = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k^{(i)}} \left\| \mathbf{x} - \mu_k^{(i)} \right\|_2^2 \qquad (8)$$

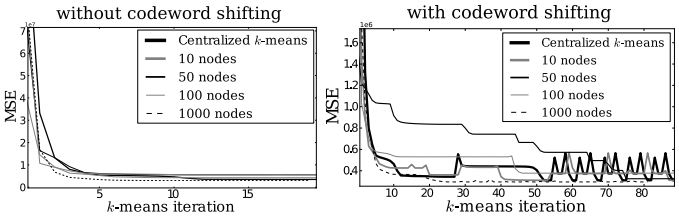

Fig. 1. Comparison of the obtained MSE_g versus a centralized K-means algorithm on a synthetic dataset.

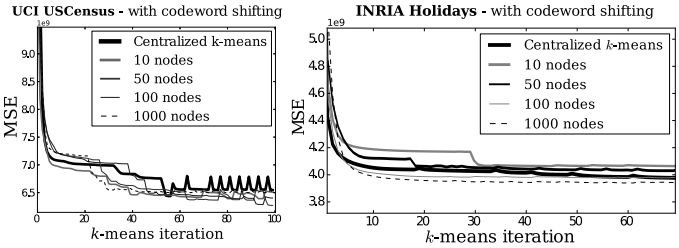

Fig. 2. MSE_g obtained on UCI UScensus90 and INRIA Holidays datasets using our algorithm versus a centralized K-means with codeword-shifting.

The relative error to consensus (REC) evaluates the consistency between nodes in terms of normalized Euclidean distance between codebooks.

$$\text{REC} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\mathcal{M}^{(i)} - \mathcal{M}^{\star}\right\|_F^2}{\left\|\mathcal{M}^{\star}\right\|_F^2} \tag{9}$$

The Percentage of Membership Mismatches (PMM) focuses on the consistency of the produced local quantizers $q^{(i)}$ with respect to the consensus one $q^{\star}$.

$$\text{PMM} = \frac{1}{N|\mathbf{X}|}\sum_{i=1}^{N}\left|\{\mathbf{x} \;/\; q^{(i)}(\mathbf{x}) \neq q^{\star}(\mathbf{x})\}\right| \tag{10}$$

A. Experiments on synthetic data

We first evaluate our algorithm on synthetic data sampled from from multiple normal laws with a very heterogeneous distribution between nodes, *i.e*, each node has its own generation law. We compare performances with a classical centralized K-means on the full data. Evaluation is made both with and without codeword-shifting. At convergence (about 30 iterations), we both achieve a very low REC (in the order of 10^{-7}) and a zero PMM. That is, all nodes get the same codebook and generate the same partitions. Moreover, Figure 1 highlights the obtained global MSE is close to the MSE given by a centralized K-means algorithm with an initial codebook set to $\mathcal{M}^{\star}(0)$. Our decentralized codeword-shifting feature brings improvements which are comparable with those obtained in a centralized setup.

B. Experiments on real-world data

Secondly, we demonstrate the scalability of our algorithm on large datasets, namely UCI Census 90 and INRIA Holidays. The former contains 2.5 millions samples with 68 attributes, and the latter is a set of 4.5 millions 128-dimensional SIFT descriptors extracted from 1491 images. Their content is spread on 10, 50, 100 and 1000 nodes, with an highly skewed distribution. Obtained MSE_g over time are reported in Figure.2. Performances in terms of MSE are comparable with a centralized K-means, while the decentralization of the processing provide much lower computational complexity. Beside, local datasets being large, the duration of the Gossip aggregation step becomes negligible with respect to the local assignment step.

C. Lowering the number of exchanged messages

Finally, we show experiments with lower values of M. Results in terms of MSE_g, PMM and REC are reported in Figure.3. Interestingly, the overall convergence of the decentralized K-means still holds even when M is very low. This means that full convergence to the consensus codebook is not really necessary, and especially that it may suffice to steadily reduce the assignment error instead of looking for a perfect match at each iteration. The slope of the PMM's decrease illustrates this cross-iterations progressive improvement. Besides, results with M higher than the theoretical bound confirm that higher values do not bring more accuracy.

VI. Conclusion

We presented a decentralized K-means clustering algorithm with codeword-shifting using a randomized Sum-Weight Gossip protocol to incrementally estimate a consensus codebook at each iteration. Nodes endowed with their local datasets iteratively converges to the same codebook, both in terms of clusters assignment and centroids locations. Classical improvements brought by codeword-shifting features in centralized settings hold when using our decentralized counter-part.

We provided a probabilistic bound on the number of messages each node has to send above which our algorithm is equivalent to a centralized K-means. The number of messages turned out to grow logarithmically with the number of nodes in the network, demonstrating the scalability of our method to large networks and datasets. This bound is backed by experiments which interestingly show that there is no need for full convergence in the decentralized aggregation step to get consistent results, suggesting that there exists a tighter bound to guarantee global consistency.

References

[1] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," in *WALCOM '09*, (Berlin, Heidelberg), pp. 274–285, Springer-Verlag, 2009.

[2] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[3] T.-C. Lu and C.-Y. Chang, "A survey of vq codebook generation," *JIHMSP*, vol. 1, no. 3, pp. 190–203, 2010.

[4] G. Patanè and M. Russo, "The enhanced LBG algorithm," *Neural Networks*, vol. 14, pp. 1219–1237, November 2001.

[5] M. Shindler, A. Meyerson, and A. Wong, "Fast and accurate k-means for large datasets," in *NIPS*, 2011.

[6] N. Ailon, R. Jaiswal, and C. Monteleoni, "Streaming k-means approximation," *NIPS*, vol. 22, pp. 10–18, 2009.

[7] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[8] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta, "Clustering distributed data streams in peer-to-peer environments," *Inf. Sci.*, vol. 176, no. 14, pp. 1952–1985, 2006.

[9] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "Dbdc: Density based distributed clustering," in *EDBT 2004*, pp. 88–105, Springer, 2004.

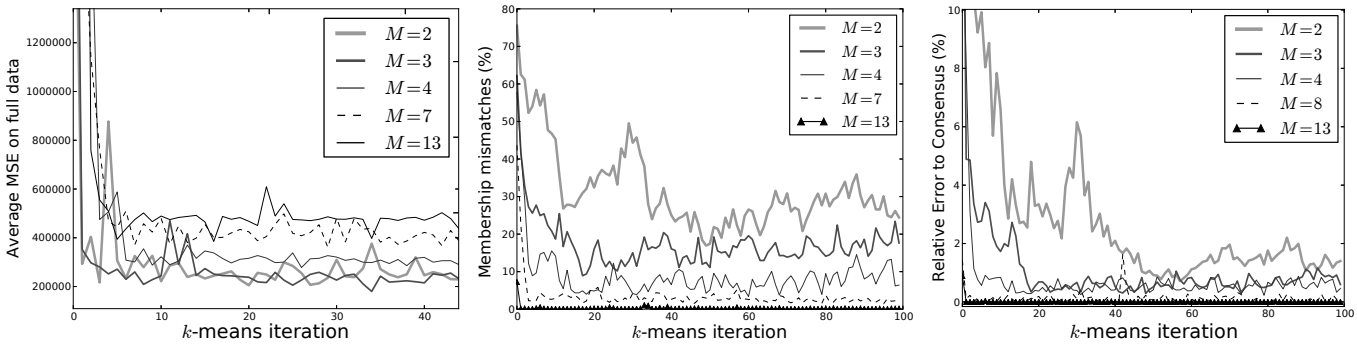

Fig. 3. MSE_g, PMM, and REC for various number of messages by node M

[10] A. Nikseresht and M. Gelgon, "Gossip-based computation of a gaussian mixture model for distributed multimedia indexing," *Multimedia, IEEE Transactions on*, vol. 10, no. 3, pp. 385–392, 2008.

[11] G. D. Fatta, F. Blasa, S. Cafiero, and G. Fortino, "Fault tolerant decentralised k-means clustering for asynchronous large-scale networks," *Journal of Parallel and Distributed Computing*, September 2012.

[12] G. D. Fatta, F. Blasa, S. Cafiero, and G. Fortino, "Epidemic k-means clustering," in *2011 IEEE 11th ICDM Workshops*, pp. 151–158, IEEE, December 2011.

[13] S. Datta, C. Giannella, and H. Kargupta, "K-means clustering over a large, dynamic network," in *SDM*, SIAM, 2006.

[14] S. Datta, C. Giannella, and H. Kargupta, "Approximate distributed k-means clustering over a peer-to-peer network," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, pp. 1372–1388, Oct. 2009.

[15] M. Durut, B. Patra, and F. Rossi, "A discussion on parallelization schemes for stochastic vector quantization algorithms," *CoRR*, vol. abs/1205.2282, 2012.

[16] M. Eisenhardt, W. Muller, and A. Henrich, "Classifying documents by distributed p2p clustering," *GI Jahrestagung*, pp. 286–291, 2003.

[17] S. Datta, C. Giannella, H. Kargupta, *et al.*, "K-means clustering over peer-to-peer networks," in *SIAM International Conference on Data Mining workshop HPDM05*, Citeseer, 2005.

[18] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *FOCS '03*, pp. 482–, IEEE, 2003.

[19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[20] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *ISIT '10*, pp. 1753–1757, IEEE, 2010.

[21] F. Iutzeler, P. Ciblat, and W. Hachem, "Analysis of sum-weight-like algorithms for averaging in wireless sensor networks," *IEEE Transactions on Signal Processing*, 2013.

APPENDIX A : PROOFS

In this section we provide the proofs for the lemmas and theorems stated in section IV. Focusing on the decentralized aggregation step at a given K-means iteration τ, we made it implicit. Therefore, $\mathcal{M}^\star(\tau)$ is simply noted $\mathcal{M}^\star$.

A. Proof of Lemma IV.1 (maximal error to satisfy zero-mismatch constraint)

Proof: Let $\epsilon_{zmc} \triangleq \max_{i,k} \left\|\mu_k^\star - \hat{\mu}_k^{(i)}\right\|_2^2$ be the maximal error on any codeword estimate over the network. Let $m_{\text{NN}}^{(i)}(\mathbf{x})$ be the m^{nth} nearest neighbor of $\mathbf{x}$ among the $\hat{\mu}_{k'}^{(i)}$ and assume ϵ_{zmc} satisfies

$$\epsilon_{zmc} < \frac{1}{2} \min_i \min_{\mathbf{x}\in\mathbf{X}_i} \left(\left\|\mathbf{x} - 2_{\text{NN}}^{(i)}(\mathbf{x})\right\|_2^2 - \left\|\mathbf{x} - 1_{\text{NN}}^{(i)}(\mathbf{x})\right\|_2^2 \right) .$$

Then for any node i and any of its local samples $\mathbf{x} \in \mathbf{X}^{(i)}$,

$$2\epsilon_{zmc} < \left\|\mathbf{x} - 2_{\text{NN}}^{(i)}(\mathbf{x})\right\|_2^2 - \left\|\mathbf{x} - 1_{\text{NN}}^{(i)}(\mathbf{x})\right\|_2^2$$

Let $l \triangleq \arg\min_k \|\mathbf{x} - \hat{\mu}_k^{(i)}\|_2^2$, *i.e*, $\hat{\mu}_l^{(i)} = 1_{\text{NN}}^{(i)}(\mathbf{x})$. Then, $\forall k \neq l, \quad \|\mathbf{x} - \hat{\mu}_k^{(i)}\|_2^2 \geq \|\mathbf{x} - 2_{\text{NN}}^{(i)}(\mathbf{x})\|_2^2$. Thus,

$$2\epsilon_{zmc} < \left\|\mathbf{x} - \hat{\mu}_k^{(i)}\right\|_2^2 - \left\|\mathbf{x} - \hat{\mu}_l^{(i)}\right\|_2^2$$
$$\left\|\mathbf{x} - \hat{\mu}_l^{(i)}\right\|_2^2 + \epsilon_{zmc} < \left\|\mathbf{x} - \hat{\mu}_k^{(i)}\right\|_2^2 - \epsilon_{zmc}$$

As $\forall k', \epsilon_{zmc} \geq \|\hat{\mu}_{k'}^{(i)} - \mu_{k'}^\star\|_2^2$,

$$\left\|\mathbf{x} - \hat{\mu}_l^{(i)}\right\|_2^2 + \left\|\hat{\mu}_l^{(i)} - \mu_l^\star\right\|_2^2 < \left\|\mathbf{x} - \hat{\mu}_k^{(i)}\right\|_2^2 - \left\|\hat{\mu}_k^{(i)} - \mu_k^\star\right\|_2^2$$

Thanks to triangle inequalities, one can note that

$$\left\|\mathbf{x} - \hat{\mu}_l^{(i)} + \hat{\mu}_l^{(i)} - \mu_l^\star\right\|_2^2 \leq \left\|\mathbf{x} - \hat{\mu}_l^{(i)}\right\|_2^2 + \left\|\hat{\mu}_l^{(i)} - \mu_l^\star\right\|_2^2$$
$$\left\|\mathbf{x} - \hat{\mu}_k^{(i)}\right\|_2^2 - \left\|\hat{\mu}_k^{(i)} - \mu_k^\star\right\|_2^2 \leq \left\|\mathbf{x} - \hat{\mu}_k^{(i)} + \hat{\mu}_k^{(i)} - \mu_k^\star\right\|_2^2$$
$$\|\mathbf{x} - \mu_l^\star\|_2^2 < \|\mathbf{x} - \mu_k^\star\|_2^2$$

This means that any node i assigns any sample $\mathbf{x} \in \mathbf{X}^{(i)}$ to the same cluster as the consensus codebook would yield. Since

$$\forall i, \quad \left\|\mathcal{M}^\star - \mathcal{M}^{(i)}\right\|_F^2 \leq K \max_k \left\|\mu_k^\star - \hat{\mu}_k^{(i)}\right\|_2^2 \leq K\epsilon_{zmc} \ ,$$

the zero-mismatch condition is satisfied provided that all nodes reach an estimation squared error lower than $K\epsilon_{zmc}$. ■

B. Proof of Lemma IV.2 (upper bound on absolute error after t exchanged messages)

Notations. Let $\hat{\mu}_{kc}^{(i)}(t)$ refers to the c^{th} component of node i's estimate for consensus codeword $\mu_k^\star$ after exchanging t messages on the network. We focus on a single locally estimated codebook entry $\hat{\mu}_{kc}^{(i)}(t)$ denoted $\hat{\mu}_i(t)$. Let $s_i(t)$ and $w_i(t)$ denote the associated sum and weight, *i.e*, $\hat{\mu}_i(t) = \frac{s_i(t)}{w_i(t)}$. Let $\mathbf{s}(t) = (s_1(t) \dots s_N(t))^T$ and $\mathbf{w}(t) = (w_1(t) \dots w_N(t))^T$. The consensus entry to estimate $\mu_{kc}^\star$ (simply noted μ) is thus given by $\frac{\|\mathbf{s}(\mathbf{0})\|_1}{\|\mathbf{w}(\mathbf{0})\|_1}$. A message event affects $\mathbf{s}(t)$ and $\mathbf{w}(t)$ by applying a diffusion matrix $\mathbf{K}(t)$:

$$\mathbf{s}(t+1)^T = \mathbf{s}(t)^T\mathbf{K}(t) \quad \text{and} \quad \mathbf{w}(t+1)^T = \mathbf{w}(t)^T\mathbf{K}(t)$$

Define $\mathbf{P}(t)$ as the product of the diffusion matrices $\mathbf{K}(1)\mathbf{K}(2)\dots\mathbf{K}(t)$.

Proof: As by definition $\hat{\mu}_j(t) = \frac{s_j(t)}{w_j(t)}$ and $\mu = \frac{\sum_l s_l(0)}{\sum_l w_l(0)}$,

$$\forall i, \quad |\hat{\mu}_i(t) - \mu|^2 \leq \sum_{j=1}^{N} |\hat{\mu}_j(t) - \mu|^2$$

$$|\hat{\mu}_i(t) - \mu|^2 = \sum_{j=1}^{N} \frac{1}{w_j(t)^2} \left| s_j(t) - w_j(t) \frac{\sum_l^N s_l(0)}{\sum_l^N w_l(0)} \right|^2$$

We know that $\mathbf{s}(t)^T = \mathbf{s}(0)^T \mathbf{P}(t)$ and $\mathbf{w}(t)^T = \mathbf{w}(0)^T \mathbf{P}(t)$. Noting $w^-(t) \triangleq \min_j w_j(t)$ yields:

$$\begin{aligned}
|\hat{\mu}_i(t) - \mu|^2 &\leq \sum_{j=1}^{N} \frac{1}{w^-(t)^2} \left| s_j(t) - w_j(t) \frac{\sum_l^N s_l(0)}{\sum_l^N w_l(0)} \right|^2 \\
&\leq \frac{1}{w^-(t)^2} \sum_{j=1}^{N} \left| \sum_{l=1}^{N} s_l(0) \mathbf{P}_{lj}(t) - \frac{\sum_l^N s_l(0).w_j(t)}{\|\mathbf{w}(0)\|_1} \right|^2 \\
&\leq \frac{1}{w^-(t)^2} \sum_{j=1}^{N} \left| \sum_{l=1}^{N} s_l(0) \left(\mathbf{P}_{lj}(t) - \frac{\sum_m^N w_m(0) \mathbf{P}_{mj}(t)}{\|\mathbf{w}(0)\|_1} \right) \right|^2 \\
&\leq \frac{\|\mathbf{s}(0)\|_2^2}{w^-(t)^2} \sum_{j=1}^{N} \left| \sum_{l=1}^{N} \left(\mathbf{P}_{lj}(t) - \frac{\sum_m^N w_m(0) \mathbf{P}_{mj}(t)}{\|\mathbf{w}(0)\|_1} \right) \right|^2 \\
&\leq \frac{\|\mathbf{s}(0)\|_2^2}{w^-(t)^2} \sum_{j=1}^{N} \sum_{l=1}^{N} \left| \mathbf{P}_{lj}(t) - \frac{\sum_m^N w_m(0) \mathbf{P}_{mj}(t)}{\|\mathbf{w}(0)\|_1} \right|^2
\end{aligned}$$

Let $\tilde{\mathbf{w}} = \mathbf{w}(0)/\|\mathbf{w}(0)\|_1$. This translates into the following Frobenius matrix norm:

$$\begin{aligned}
|\hat{\mu}_i(t) - \mu|^2 &\leq \frac{\|\mathbf{s}(0)\|_2^2}{w^-(t)^2} \left\| \mathbf{P}(t) - \mathbf{1}\tilde{\mathbf{w}}^T \mathbf{P}(t) \right\|_F^2 \\
&\leq \frac{\|\mathbf{s}(0)\|_2^2}{w^-(t)^2} \left\| \left(\mathbf{I} - \mathbf{1}\tilde{\mathbf{w}}^T \right) \mathbf{P}(t) \right\|_F^2
\end{aligned}$$

We now consider this result for any codebook component μ_{kc}. Remembering that $\tilde{\mathbf{w}}$ depends on k, we note this vector $\tilde{\mathbf{w}}_k$:

$$\tilde{\mathbf{w}}_k \triangleq \frac{1}{\sum_i w_k^{(i)}(0)} (w_k^{(1)}(0) \dots w_k^{(N)}(0))^T \ .$$

Let $\mathbf{J}_k = \mathbf{I} - \mathbf{1}\tilde{\mathbf{w}}_k^T$ and $\psi_k(t) = \|\mathbf{J}_k \mathbf{P}(t)\|_F^2$. We get an upper bound on the estimation error for any codebook component:

$$\forall i, k, c, \quad \left| \hat{\mu}_{kc}^{(i)}(t) - \mu_{kc}^{\star} \right|^2 \leq \frac{\sum_i s_{kc}^{(i)}(0)^2}{\min_i w_k^{(i)}(t)^2} \psi_k(t)$$

By summing for all components, we bound the codebooks estimation error at any node i:

$$\forall i, \quad \left\| \hat{\mathcal{M}}^{(i)}(t) - \mathcal{M}^{\star} \right\|_F^2 \leq KD \max_k \frac{\max_c \sum_i s_{kc}^{(i)}(0)^2}{\min_i w_k^{(i)}(t)^2} \psi(t)$$
$$\text{where} \quad \psi(t) = \max_k \psi_k(t)$$

Without loss of generality, we can consider that all data is normalized inside the unit sphere. This leads to $\max_c \sum_i s_{kc}^{(i)}(0)^2 \leq \sum_i w_k^{(i)}(0)^2$, which gives our claimed result. ■

C. Proof of Lemma IV.3 (upper bound on $\psi(t)$)

Proof: We first express the conditional expectation of $\psi_k(t+1)$ given $\psi_k(t)$ for any k. We use the fact that $\forall \mathbf{A}, \|\mathbf{A}\|_F^2 = \text{Tr}(\mathbf{A}\mathbf{A}^T)$:

$$\begin{aligned}
\forall k, \mathbb{E}\left[\psi_k(t+1) \middle| \psi_k(t)\right] &= \mathbb{E}[\text{Tr}(\mathbf{J}_k \mathbf{P}(t+1)\mathbf{P}(t+1)^T \mathbf{J}_k^T) | \psi_k(t)] \\
&= \text{Tr}(\mathbf{J}_k \mathbf{P}(t) \mathbb{E}\left[\mathbf{K}(t+1)\mathbf{K}(t+1)^T \middle| \psi_k(t)\right] \mathbf{P}(t)^T \mathbf{J}_k^T)
\end{aligned}$$

Since $\mathbf{K}(t)$ is stationary and does not depend on $\psi_k(t-1)$,

$$\mathbb{E}\left[\psi_k(t+1) | \psi_k(t)\right] = \text{Trace}(\mathbf{J}_k \mathbf{P}(t) \mathbb{E}[\mathbf{K}\mathbf{K}^T] \mathbf{P}(t)^T \mathbf{J}_k^T)$$

We further calculate $\mathbb{E}\left[\mathbf{K}\mathbf{K}^T\right]$ using the distribution of the diffusion matrices $\mathbf{K}(t)$ defined in Eq.7:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{K}\mathbf{K}^T\right] &= \frac{1}{N(N-1)} \sum_{i,j \neq i} \left(\mathbf{I} - \frac{1}{2} \mathbf{e}_i \left(\mathbf{e}_i - \mathbf{e}_j \right)^T \right) \left(\mathbf{I} - \frac{1}{2} \left(\mathbf{e}_i - \mathbf{e}_j \right) \mathbf{e}_i^T \right) \\
&= \mathbf{I} - \frac{1}{2N(N-1)} \sum_{i,j \neq i} \left(2\mathbf{e}_i \mathbf{e}_i^T - \mathbf{e}_j \mathbf{e}_i^T - \mathbf{e}_i \mathbf{e}_j^T - \mathbf{e}_i \mathbf{e}_i^T \right) \\
&= \mathbf{I} - \frac{1}{2N(N-1)} \left((N-1)\mathbf{I} - \mathbf{1}\mathbf{1}^T + \mathbf{I} - \mathbf{1}\mathbf{1}^T + \mathbf{I} \right) \\
&= \left(1 - \frac{N+1}{2N(N-1)} \right) \mathbf{I} + \frac{1}{N(N-1)} \mathbf{1}\mathbf{1}^T
\end{aligned}$$

Consequently, $\mathbb{E}[\psi_k(t+1) | \psi_k(t)] =$

$$\text{Tr}\left[\mathbf{J}_k \mathbf{P}(t) \left((1 - \frac{N+1}{2N(N-1)}) \mathbf{I} - \frac{1}{N(N-1)} \mathbf{1}\mathbf{1}^T \right) \mathbf{P}(t)^T \mathbf{J}_k^T \right]$$

As $\mathbf{K}(t)\mathbf{1} = \mathbf{1}$ and $\mathbf{J}_k \mathbf{1}\mathbf{1}^T = \mathbf{0}$, $\mathbf{J}_k \mathbf{P}(t) \mathbf{1}\mathbf{1}^T = \mathbf{0}$. Hence,

$$\begin{aligned}
\mathbb{E}[\psi_k(t+1) | \psi_k(t)] &= \left(1 - \frac{N+1}{2N(N-1)} \right) \text{Tr}(\mathbf{J}_k \mathbf{P}(t) \mathbf{P}(t)^T \mathbf{J}_k^T) \\
&= \left(1 - \frac{N+1}{2N(N-1)} \right) \psi_k(t)
\end{aligned}$$

From this recursion, and since $\psi_k(0) = \text{Tr}(\mathbf{J}_k \mathbf{J}_k^T) = \|\tilde{\mathbf{w}}_k\|_2^2 \leq N$, we get the following bound on $\mathbb{E}[\psi(t)]$:

$$\mathbb{E}[\psi(t)] \leq N \left(1 - \frac{N+1}{2N(N-1)} \right)^t \tag{11}$$

For a given constant ϵ, we want $\mathbb{P}[\psi(t) \leq \epsilon]$ to be greater than a given parameter δ. Thanks to Markov's inequality, we know that

$$\forall t, \forall \epsilon, \quad \mathbb{P}[\psi(t) \geq \epsilon] \leq \frac{\mathbb{E}[\psi(t)]}{\epsilon}$$

We thus look for values of t such that $\frac{\mathbb{E}[\psi(t)]}{\epsilon}$ is lower than $1 - \delta$. Using Eq.11, it is sufficient that

$$\begin{aligned}
1 - \delta &\geq \frac{N}{\epsilon} \left(1 - \frac{N+1}{2N(N-1)} \right)^t \\
\ln(1-\delta) &\geq \ln(N) - \ln(\epsilon) + t \ln\left(1 - \frac{N+1}{2N(N-1)} \right) \\
t &\geq \frac{\ln N - \ln(1-\delta) - \ln \epsilon}{\ln(1 - \frac{N+1}{2N(N-1)})^{-1}}
\end{aligned}$$

This proves the bound on $\psi(t)$ claimed in Lemma IV.3. ■

D. Proof of Lemma IV.4 (lower bound on weights)

Proof: After any number t of exchanged messages, it is obvious that for any value of k there is always at least one node, say z_t, whose weight $w_k^{(z_t)}$ is greater or equal to the average $\bar{w}_k$ of all nodes weights for k. Mass conservation law ensures that $\bar{w}$ doesn't depend on t:

$$\forall t, \forall k, \exists z_t, \quad w_k^{(z_t)}(t) \geq \bar{w}_k \triangleq \frac{1}{N}\sum_i w_k^{(i)}(t)$$

Weights decrease only when a message is sent, and according to our definition of $\mathbf{K}(t)$, the weights of the sender are divided by 2 and others increase or remain constant. Therefore, as receivers add incoming weights to their current ones, if $w_k^{(z_t)}$ diffuses to a node i before $t+\Delta t$, $w_k^{(i)}(t+\Delta t)$ will be at least $\bar{w}_k 2^{-\Delta t}$:

$$\mathbf{P}_{z_t,i}(t+\Delta t)\neq 0 \Rightarrow \mathbf{P}_{z_t,i}(t+\Delta t) \geq 2^{-\Delta t} \Rightarrow w_k^{(i)}(t+\Delta t) \geq \bar{w}_k 2^{-\Delta t}$$

We study how $w_k^{(z_t)}$ diffuses over the network, *i.e*, the number Δt of messages required to have information diffused from z_t to all other nodes during $[t, t+\Delta t]$. As the distribution of the $\mathbf{K}(t)$ is not time-dependent, this amounts to finding Δt such that all entries in row z_t of $\mathbf{P}(\Delta t)$ are strictly positive. We proceed by counting the number $R_z(t)$ of such positive entries in any given row z. As $\mathbf{P}(0) = \mathbf{I}$, $R_z(0) = 1$. When the t^{nth} message is sent from node i to node j, $R_z(t)$ increases by 1 only if i had a non-null entry and j had a null entry. The pair (i, j) being uniformly drawn among $N(N-1)$ possible message events, the probability that R increases depends solely on its current value:

$$\forall t > 0, p_r \triangleq \mathbb{P}\left[R(t+1) - R(t) = 1 \mid R(t) = r\right] = \frac{r(N-r)}{N(N-1)}$$

This defines a pure-birth stochastic process with a probability p_r of transition from state r to state $r+1$ quadratic in r. We are then interested in the probability $P(t)$ to reach state N in t steps, starting from state 1, which is given by the entry $1, N$ of the transition matrix raised to power t:

$$P(t) \triangleq \mathbb{P}[R(t) = N \mid R(0) = 1] = (\mathbf{M}^t)_{1,N}$$

Conversely, given a parameter δ, we obtain the number of exchanged messages required to get diffusion of $w_k^{(z_t)}$ over the whole network with probability at least δ, by computing the value T_δ for which $\forall t > T_\delta, P(t) \geq \delta$. Empirical values for T_δ are given in Appendix B. This yields the following lower bound on weights:

$$\forall k, \forall \delta, \forall t > T_\delta, \quad \mathbb{P}[\min_i w_k^{(i)}(t) \geq \bar{w}_k 2^{-T_\delta}] \geq \delta$$

■

E. Proof of Theorem IV.5 (number of exchanged messages for zero-mismatch)

Proof: Using Lemma IV.1, we know that the zero mismatch condition is satisfied provided that

$$\forall i, \quad \|\mathcal{M}^{(i)}(t) - \mathcal{M}^\star\|_F^2 \leq K\epsilon_{zmc}$$

Using Lemma IV.2 then Lemma IV.4, a sufficient condition is

$$KD\max_k \frac{\sum_i w_k^{(i)}(0)^2}{\min_i w_k^{(i)}(t)^2}\psi(t) \leq K\epsilon_{zmc}$$

$$\forall k, \forall \delta_1, \forall t > T_{\delta_1}, \quad \mathbb{P}\left[\min_i w_k^{(i)}(t) \geq \bar{w}_k.2^{-T_{\delta_1}}\right] \geq \delta_1$$

Then when $t > T_{\delta_1}$, we have with probability greater than δ_1

$$\begin{aligned}\forall k, \quad \frac{\sum_i w_k^{(i)}(0)^2}{\min_i w_k^{(i)}(t)^2} &\leq N^2 2^{2T_{\delta_1}} \frac{\sum_i w_k^{(i)}(0)^2}{(\sum_i w_k^{(i)}(0))^2} \\ &\leq N^2 2^{2T_{\delta_1}} \frac{(\sum_i w_k^{(i)}(0))^2}{(\sum_i w_k^{(i)}(0))^2} = N^2 2^{2T_{\delta_1}}\end{aligned}$$

Thus it is sufficient that $DN^2 2^{2T_{\delta_1}}\psi(t) \leq \epsilon_{zmc}$, thus

$$\psi(t) \leq \frac{\epsilon_{zmc}}{N^2 2^{2T_{\delta_1}} D}$$

By Lemma IV.3, this is guaranteed with probability greater than δ_2 when

$$\begin{aligned}t &\geq \frac{\ln N - \ln(1-\delta_2) - \ln \epsilon_{\text{zmc}} + 2\ln N + 2\ln 2T_{\delta_1} + \ln D}{\ln \frac{2N^2-2N}{2N^2-3N-1}} \\ &\geq \frac{3\ln N - \ln(1-\delta_2) - \ln \epsilon_{\text{zmc}} + 2\ln 2T_{\delta_1} + \ln D}{\ln \frac{2N^2-2N}{2N^2-3N-1}} = \mathcal{T}\end{aligned}$$

Remarking that $\forall N, 2\ln 2 \geq \ln \frac{2N^2-2N}{2N^2-3N-1}$, we have $\mathcal{T} \geq T_{\delta_1}$. To conclude, provided that $t \geq \mathcal{T}$,

$$\begin{aligned}&\mathbb{P}\left[\min_i w_i(t) \geq \bar{w} 2^{-T(\delta_1)}\right] \geq \delta_1 \text{ and} \\ &\mathbb{P}\left[\psi(t) \leq \frac{\epsilon_{zmc}}{N^2 2^{2T_{\delta_1}} D} \mid \min_i w_i(t) \geq \bar{w} 2^{-T(\delta_1)}\right] \geq \delta_2 \\ \Rightarrow &\mathbb{P}\left[\psi(t) \leq \frac{\epsilon_{zmc}}{N^2 2^{2T_{\delta_1}} D} \cap \min_i w_i(t) \geq \bar{w} 2^{-T(\delta_1)}\right] \geq \delta_1\delta_2\end{aligned}$$

Hence, given a parameter δ, and taking $\delta_1 = \delta_2 = \sqrt{\delta}$, we get our final bound on the number of messages to exchange to meet the zero-mismatch constraint with probability at least δ:

$$\forall \delta, \exists \mathcal{T}, \forall t \geq \mathcal{T}, \quad \mathbb{P}\left[\forall i, \|\hat{\mathcal{M}}^{(i)} - \mathcal{M}^\star\|_F^2 \leq K\epsilon_{zmc}\right] \geq \delta$$

where

$$\mathcal{T} = \frac{3\ln N + 2\ln 2T_{\delta^{1/2}} + \ln D - \ln(1-\delta^{1/2}) - \ln \epsilon_{zmc}}{\ln \frac{2N^2-2N}{2N^2-3N-1}}$$

■

APPENDIX B : PRACTICAL VALUES FOR T_δ

To compute practical values for T_δ, we use a lookup table. T_δ versus δ and N is shown in Figure.4 below:

T_δ	$\delta = 0.2$	0.5	0.8	0.9	0.99
$N = 10$	37	49	64	74	101
30	184	223	271	301	387
60	459	537	634	694	867
100	872	1003	1165	1265	1554
140	1319	1502	1728	1869	2274

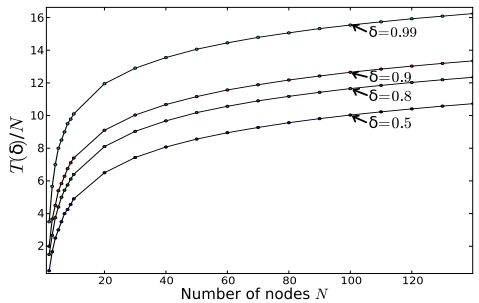

Fig. 4. Diffusion time T_δ versus δ and the number of nodes N.